Chapter 3: Logging into UNIX Systems at Fermilab

This chapter is intended to show you how to log into a Kerberized UNIX computer at Fermilab, at the console or over the network.

Much of the material in this chapter has been reproduced from the **Strong Authentication at Fermilab** manual, *Chapter 4: Accessing Kerberized Machines (Fermilab-Supported Methods)* and edited appropriately.

3.1 Overview

3.1.1 About Logging in and Authenticating to Kerberos

Logging into a UNIX machine and authenticating to Kerberos are separate functions, but they are linked. Your Kerberized UNIX desktop machine is configured either to use the Kerberized login program or a standard UNIX one. In the former case, you log in and authenticate in one step. In the latter case, you log in first, then authenticate manually by running a command called **`kinit`**.

Connecting from your Kerberized desktop to remote Kerberized machines requires that you authenticate locally, making sure to obtain forwardable credentials (credentials are also called “tickets”), and that you forward these credentials to the target machine as you log into it. This requires use of Kerberized connection programs, e.g., Kerberized versions of telnet, ssh, ftp, and so on. You should never authenticate directly on a remote host; this would expose your Kerberos password on the network.

That said, here is an exception! If your desktop is unKerberized, you can’t authenticate to Kerberos from it. Authenticating to Kerberos does not become an issue until you need to connect over the network to a remote Kerberized host. For this you will need a CRYPTOCard (described in section 3.5 *Connecting from a NonKerberized Machine to Kerberized Host*). When you log onto a remote host using a CRYPTOCard, you authenticate directly on the remote machine. But it happens safely; your Kerberos password does not pass over the network.

☞ Note that if you have an account and a standard UNIX (or AFS) password (in the `passwd` file or NIS map) on a machine, but no Kerberos principal or password, you can log in LOCALLY TO THE MACHINE ONLY. From any terminal other than the console, the Kerberized machine responds in portal mode (described in section 3.5 *Connecting from a NonKerberized Machine to Kerberized Host*) and you are given no option to enter your UNIX password.

3.1.2 About Forwarding Kerberos Tickets

In order to forward your Kerberos credentials to a remote host, the credentials must be forwardable. Whether your credentials are forwardable or not depends upon the default settings configured on your system, and/or on the options you use with the **`kinit`** command, if you run that manually. Here’s what you need to know:

After you’ve authenticated, run the command

```
% klist -f
```

Look for the **`F`** flag in the output (third line, below):

```
Valid starting    Expires          Service principal
02/11/00 12:45:33 02/12/00 01:45:33 krbtgt/FNAL.GOV@FNAL.GOV
       Flags: FIA
```

If it’s there, your credentials are forwardable. If not, you’ll need to reauthenticate using **`kinit`** with the **`-f`** option (make sure it’s lowercase **`f`**!):

```
% kinit -f
```

💣 Do not run **`kinit`** over the network to authenticate on a remote machine. **`kinit`** requires entry of your Kerberos password, and it is against Fermilab policy to send your Kerberos password over the network, even encrypted. As of Kerberos v1_5, **`kinit`** is equipped with a warning that appears if the userid issuing the command doesn’t own the console device. It is designed to help users avoid typing their password inadvertently over the network.

Forwarding is described in the **Strong Authentication at Fermilab** manual section *9.2.4 Forwarding Tickets*.

3.2 Logging In at the Console of a Kerberized UNIX Machine

3.2.1 Using Standard UNIX Login Program

If your desktop machine is running the standard UNIX login program, log in at the console normally, by entering your login id and your standard UNIX password. The standard login program does not accept your Kerberos password.

AFS

Note that if your machine runs AFS, your UNIX and AFS passwords may be the same.

To obtain your Kerberos credentials after you log in, run:

```
% kinit <-f> <principal_name>
```

The `-f` flag requests forwardable tickets; forwarding may be set “on” by default for your system in which case the `-f` is unnecessary. (Run `kinit` without `-f`, then run `klist -f` to see if the `F` flag is present.) Only include your principal name if it is different from your login id on the machine you’re using. Your credentials should get forwarded to other strengthened machines normally, as you connect to them using Kerberized services.

3.2.2 Using Kerberos Login Program

You will authenticate to Kerberos when you log in to your desktop if:

- your machine is configured to use the **kerberos** login program,[1]
- your login id on the machine matches your Kerberos principal, and
- you enter your Kerberos password at the password prompt.

Just enter your login id and Kerberos password as prompted. You do not need to run `kinit` after login, unless you want to supply options that would give your credentials non-default settings.

An advantage to using the **kerberos** login program is that it checks the configuration file in which you or your system administrator can set defaults for Kerberized applications. For more information on this, see the **Strong Authentication at Fermilab** document, *Chapter 17: The Kerberos Configuration File: krb5.conf*.

[1] Not applicable to IRIX systems, or to Linux and Solaris if using the GUI login box unless you’re running vendor-supplied Kerberos login PAMs.

3.2.3 If you don’t have a principal yet...

☞ Note that if you have an account and a standard UNIX password on a machine (in the `passwd` file or NIS map) but no principal or Kerberos password, you can still log in at the console. Just enter your login id and UNIX (or AFS) password as prompted. (From any terminal other than the console, the Kerberized machine looks for existing Kerberos credentials, and responds in portal mode if none are found; you have no opportunity to enter your UNIX password.) However, once logged in, you cannot make outbound connections to Kerberized hosts from there since Kerberized services are unavailable to you.

3.3 Connecting from One Kerberized Machine to Another

Make sure you have forwardable credentials on your desktop machine (see section 3.1.2 *About Forwarding Kerberos Tickets*), then run the Kerberized version of the connection program you want to use (**ssh**, **slogin**, **telnet**, **rsh**, **rlogin**, **rcp**, **scp** or **ftp**) to connect and forward your credentials to the target machine.

If your machine is configured properly, the Kerberized versions of these programs should be the default versions. If you’re not sure, run **`which <program>`**, and check the output for the path shown below, e.g.,:

```
% which telnet
   /usr/krb5/bin/telnet
```

This is the right path for all the connection programs. To run Kerberized telnet in this case, just type:

```
% telnet <-f> <-l remote_login_id> host
```

Use the **`-f`** option if your Kerberized telnet does not forward credentials by default. If your login ids are different on the “source” and “target” hosts, use the **`-l`** option with your target host login id as the argument.

If the **`which`** output shows a different path than the one shown above, the Kerberized version of the program is not the default. In this case, specify the Kerberized program by using the full path, e.g.,:

```
% /usr/krb5/bin/telnet <-f> <-l remote_login_id> host
```

The Kerberized features of these programs are described in the **Strong Authentication at Fermilab** manual’s *Chapter 13: Network Programs Available on Kerberized Machines*.

Assuming your credentials get forwarded to the target machine, you should be automatically recognized and authenticated there; you should not be prompted for your Kerberos password.

A few notes:

- If the usernames on the machines differ, use the `-l <remote_login_id>` option; e.g., `ssh -l <remote_login_id>`.
- If ticket forwarding has been set "off" for your system, and you want to connect to a Kerberized machine with ticket forwarding turned on, use the appropriate option, e.g., `-f` or `-F` for **telnet**, **rsh**, and **rlogin** (`-F` marks them reforwardable from the target machine to other machines, whereas `-f` does not).
- If ticket forwarding has been set "on" for your system, and you want to connect to a Kerberized machine with ticket forwarding turned off, use the appropriate option (e.g., `-N` for **telnet**, **rsh**, **rlogin**, and **rcp**, or `-k` for Kerberized **ssh**).

3.4 Connecting from Off-Site

For this topic we refer you to the **Strong Authentication at Fermilab** guide *Chapter 6: Logging In from Off-Site*.

3.5 Connecting from a NonKerberized Machine to Kerberized Host

3.5.1 About Portal Mode

If your local desktop computer does not run Kerberos software and is not configured for the FNAL.GOV realm, then you can't authenticate to FNAL.GOV locally on your computer. You can work on the desktop with no problem, but in order to connect over the network to Kerberized UNIX hosts, you must authenticate to FNAL.GOV first.

Kerberized machines in the FNAL.GOV realm are configured to require entry of a single-use password whenever they receive a login request coming from an unKerberized computer over the network. (The password gets transmitted over the network, and it could get intercepted. That's why it must become invalid after one login.) The target computer is said to respond in *portal mode* in this case. It is acting as a secure gateway into the strengthened realm.

How do you get a single-use password that Kerberos will recognize and honor? The FNAL.GOV realm at Fermilab is setup to use CRYPTOCards to provide these single-use passwords.

Once you've logged on successfully through the portal, the KDC "knows who you are", and the machine obtains your Kerberos credentials for you. You are not required to provide your Kerberos password when making further network connections to other machines in the FNAL.GOV realm. If you need to reauthenticate (which you have to do on the remote host in this case!), run the command **`new-portal-ticket`**. This provides a portal mode prompt, and is safe to run over the network.

3.5.2 Log in via CRYPTOCard

The CRYPTOCard login code assumes that your login name on the Kerberized machine matches your principal. If your names don't match, you won't be able to log in using this method.

Logins to Kerberized machines via CRYPTOCard can be done from your non-Kerberized machine using the standard, non-Kerberized **`telnet`**, **`FTP`**, or (if **ssh** v1_2_27d or later is installed on target machine) non-Kerberized **`ssh`** or **`slogin`**. The Kerberized versions of these programs are not available on non-Kerberized machines.

`rlogin`, **`rsh`** and **`rcp`** are not available for portal mode. Regarding **ssh**, you cannot use **`scp`** or **`ssh <host> <command>`** via CRYPTOCard; only interactive **`ssh`** or **`slogin`** work.

The system will prompt you to provide a non-reusable password (also known as a *response* to a *challenge*). The prompt looks like this:

```
Press ENTER and compare this challenge to the one on your
display: [12345678]
Enter the displayed response:
```

Do not enter your Kerberos password! Use your CRYPTOCard to generate a response (a single-use password). This is described in **Strong Authentication at Fermilab** *Chapter 5: Using your CRYPTOCard*.

3.6 Logging Out

Before logging out, we recommend that you destroy your Kerberos credentials to avoid a security risk. To do this, enter:

```
% kdestroy
```

If you are on a system that runs AFS, this will destroy your AFS token, too.

The command for logging out that works universally is:

```
% exit
```

The C shell family also supports:

```
% logout
```

The control character **eof**, which is usually set to **<CTRL-D>**, is the “normal” UNIX way of logging off. However, since this is easy to enter accidentally, FullFUE (described in section 1.3 *The Fermi UNIX Environment (FUE) and Product Support*) includes the command **`set ignoreeof`** in the login files to disable it (**`ignoreeof`** is not supported in all Bourne family shells, and so **<CTRL-D>** works in some of them).

If you have other processes running you will be informed that you have stopped jobs. You can continue to enter **`exit`** or **`logout`** until all the processes are terminated.

Some shells automatically look for a logout script in your home directory and run it if found. (FUE does not provide this file.) You can create a logout script that contains the above and/or other commands and use it in place of the standard logout commands. For the C shell family, call the script `.logout`. For bash, use `.bash_logout`. This isn’t available for sh and ksh.